

Algoritmos y Estructuras de Datos II

Segundo parcial – 10 de Junio de 2022

Aclaraciones

- El parcial es a libro casi-cerrado. Solo es posible tener impreso el teorema maestro y el apunte de *divide y vencerás*. *Además*, pueden tener una hoja (2 carillas), escrita a mano, con los apuntes que se deseen.
- Cada ejercicio debe entregarse en **hojas separadas**. Las mismas deben estar numeradas.
- Incluir en esta hoja: nombre, apellido, el número de orden asignado, número de libreta.
- Cada ejercicio se calificará con Perfecto, Aprobado, Regular, o Insuficiente.
- El parcial estará aprobado si las notas de los tres ejercicios tienen al menos dos A.
- Los ejercicios **no** se recuperan por separado.
- Se encuentran disponibles para utilizar todas las estructuras de datos presentadas en la *teórica con las operaciones* y complejidades dadas en las mismas.

Ej. 1.

Proponer un algoritmo que permita ordenar un arreglo de naturales (sobre los que no se conoce nada en especial) de manera ascendente en $O(n \log d)$ en el peor caso, donde n es la cantidad de elementos a ordenar y d es la cantidad de elementos distintos. Justifique informalmente por qué su algoritmo cumple con lo pedido.

Ej. 2.

Se tiene una secuencia de alturas h_1, \dots, h_n . Decimos que un intervalo $h_t, h_{t+1}, \dots, h_{t+k}$ es un *edificio* si para todo $i \in [t, t+k]$, el valor $|h_i - h_{i+1}|$ no es mayor que una cierta tolerancia dada θ y además tanto $|h_t - h_{t-1}|$ como $|h_{t+k} - h_{t+k+1}|$ (en caso de existir) son mayores que θ . El *ancho* de un edificio es la cantidad de alturas que lo componen. Por ejemplo, si la secuencia de alturas es

[100, 101, 100, 103, 80, 77, 74, 200, 32, 30]

y $\theta = 3$, entonces los edificios de esta secuencia de alturas serían [100, 101, 100, 103], [80, 77, 74], [200], [32, 30], y sus anchos serían 4, 3, 1 y 2, respectivamente.

Escribir un algoritmo que tome un arreglo de enteros (que representan alturas) y una tolerancia θ y devuelva las mismas ordenadas. El orden estará dado según los anchos de los edificios en forma creciente. Tanto los edificios como las alturas dentro de cada edificio deben mantener el orden original. En el ejemplo anterior, el resultado esperado sería [200, 32, 30, 80, 77, 74, 100, 101, 100, 103]. La complejidad del algoritmo no debe ser peor que $O(n)$, donde n es la cantidad de alturas dada. Justifique informalmente la correctitud del algoritmo y su complejidad temporal.

Ej. 3.

Se tiene un arreglo ordenado de n números naturales consecutivos con k huecos, es decir, en el arreglo están presentes todos los elementos dentro de un rango determinado salvo una cantidad k de ellos. Por ejemplo, el arreglo $A = [11, 12, 13, 15, 16, 19, 20, 21]$ tiene huecos en los valores [14, 17, 18]. Se pide describir un algoritmo que devuelva una lista con todos los huecos de un arreglo. Se puede asumir que el arreglo tiene tamaño potencia de 2.

- a) Dar un algoritmo que use la técnica de *Divide and Conquer* y resuelva el problema en tiempo $O(k \log(n))$ en el peor caso.
- b) Marcar claramente qué partes del algoritmo se corresponden a *dividir*, *conquistar* y *unir* subproblemas.
- c) Asumiendo que $k = 1$, justificar formalmente que el algoritmo cumple con la complejidad pedida.
- d) Justificar informalmente, para el caso general $k > 1$, que la complejidad es $O(k \log(n))$.